



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/741,219	12/19/2000	Adam Bosworth	104402.00119	7676
74739 7590 11/12/2010 Squire, Sanders & Dempsey L.L.P. Oracle International Corporation 8000 Towers Crescent Drive 14th Floor Vienna, VA 22182				
EXAMINER				
VU, TUAN A				
ART UNIT		PAPER NUMBER		
2193				
NOTIFICATION DATE		DELIVERY MODE		
11/12/2010		ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ipgeneraltyc@ssd.com
bgoldsmith@ssd.com
smcvean@ssd.com



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/741,219
Filing Date: December 19, 2000
Appellant(s): BOSWORTH ET AL.

Keith Mullervy
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 8/27/2010 appealing from the Office action mailed 7/13/2010.

(1) Real Party in Interest

The examiner has no comment on the statement, or lack of statement, identifying by name the real party in interest in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The following is a list of claims that are rejected and pending in the application:

Claims 1, 5-11, and 15-21.

(4) Status of Amendments After Final

The examiner has no comment on the appellant's statement of the status of amendments after final rejection contained in the brief.

(5) Summary of Claimed Subject Matter

The examiner has no comment on the summary of claimed subject matter contained in the brief.

(6) Grounds of Rejection to be Reviewed on Appeal

The examiner has no comment on the appellant's statement of the grounds of rejection to be reviewed on appeal. Every ground of rejection set forth in the Office action from which the appeal is taken (as modified by any advisory actions) is being maintained by the examiner except for the grounds of rejection (if any) listed under the subheading "WITHDRAWN

REJECTIONS.” New grounds of rejection (if any) are provided under the subheading “NEW GROUNDS OF REJECTION.”

(7) Claims Appendix

The examiner has no comment on the copy of the appealed claims contained in the Appendix to the appellant’s brief.

(8) Evidence Relied Upon

W3C Recommendation, Version 1.0 - 16 November 1999 with ‘XML Path Language (Xpath)’ pp 1-32 or <<http://www.w3.org/TR/1999/REC-xpath-19991116>>; and “XSL Transformations (XSLT)” pp. 1-92 (herein W3C) or <<http://www.w3.org/TR/xslt>>

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

/*****/

Claims Rejections – 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1, 5-11, 15-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over *W3C Recommendation*, Version 1.0; 16 November 1999, including ‘XML Path Language (Xpath)’ (herein W3C-Xpath) at <<http://www.w3.org/TR/1999/REC-xpath-19991116>>, and ‘XSL Transformation (XSLT)’ (herein W3C-Xslt or **W3C**) at <<http://www.w3.org/TR/xslt>>

As per claim 1, W3C (or W3C-Xslt) discloses a computer-implemented method of cell-based data processing that facilitates the execution of computer programming code by a computer system, the method comprising:

receiving as input computer code a data processing specification comprising a plurality of cells (e.g. W3C: source tree, set of template rules, template instantiated for a particular source element – Introduction 1 - pg. 4 – Note: XSLT processor treating a source document as a tree of nodes with mapping node with template reads on receiving data specification specifications having cells – see sec. 3: Data model; sec. 5 pg. 18), wherein each cell comprises a formula specifying an action or computation to perform (e.g. W3C: templates rules, *xsl: template match* = ... *xsl:apply-template select= ...>* – sec. 5.4 pg 21-22; sec. 7.6.1, pg. 36) when the cell is executed, and one or more attributes referencing other cells (W3C: sec. 11.2—11. 6, pg. 50-51), wherein the formula of a first cell may reference a value of a second cell (e.g. *<xsl:template match = “person”... value-of select=@given-name; <xsl: template match= person ... value-of select=“given name”* – sec. 7.6.1, 7.6.2 pg. 36-37);

wherein each cell is delineated by a beginning and ending tag (e.g. sec. 7.6.1 pg. 36; sec 11.6, pg. 51-52, bottom), wherein one of the cells is reserved as an output cell for outputting a result of the processing (*xsl: output, xsl: output method* – pg. 7; chp. 16.1, 16.2 pg. 64-68; *xsl:output* pg. 75);

parsing the specification to determine an interdependency of the plurality of cells and generating and storing a graph of the interdependency as an execution flow (e.g. *source tree* – Introduction 1, pg. 4 – Note: tree processing – see *match pattern... source nodes to which the rule applies, processes its immediate children, processed in document order* - sec. 5.3, 5.4 pg.

21-22 respecting order of instantiating templates from analysis of source tree to create a result tree via matching one node source into a template rule reads on parsing and generating graph of interdependency – bottom pg. 4; sec. 2.1-2.5 pg. 6-10); and

executing the specification in accordance with the execution flow, wherein the executing comprises evaluating the formula of each cell in the execution flow (e.g. sec. 7.6.1, 7.6.2 pg. 36-37; sec. 2.1-2.5 pg. 4-10; sec. 5.3, 5.4 – Note: processing match and analyzing node dependency and pattern correctness from source to generate proper template constructs leading to a output tree **reads on** executing in accordance with execution flow of tree source) and generating an output result (result tree – Introduction 1 pg. 4; sec. 11.1 pg. 48; *result tree* - sec. 7 pg.27-34).

wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell (sec. 7.6.1, 7.6.2 pg. 36-37 - Note: computing a value - value of select="given name" -- in one cell so that result is referenced – see @given name - in another cell reads on interlocked aspect between cells related by the formula – value-of select)

W3C does not explicitly disclose graph of interdependency as a directed graph. W3C discloses parsing to determine nodes in terms of their attributes, syntax, content relationship with respect to forward compatibility among style-sheet descendant with respect to top level (see W3C: sec. 2.2 →2.5 pg. 8-12), and for a list of source nodes, creating result tree has to be based on building in the order of the source nodes (e.g. list of source nodes appending each member of the list in order – sec 5.1, pg. 18), thus, the concept of traversing a tree with direction is conveyed. In view of such tree directional flow analysis or ordered result tree building, it would have been obvious for one skill in the art at the time the invention was made to implement the “source tree” (or graph of interdependency) as supporting the analysis by W3C so that the

generated tree implicate a directional layout of interdependency of nodes represented source document, because traversing the dependency tree in such direction would support the above syntactic compatibility of children with respect to a upper layer node, compatibility needed to be resolved prior to generating a result tree.

As per claim 5, W3C discloses wherein the first cell has a first attribute referencing a second attribute of said second cell (sec 11.6, pg. 51-52, bottom; *template match ... select value-of* - sec. 7.6.1, 7.6.2 pg. 36-37; sec 7.7, pg. 38).

As per claim 6, W3C discloses wherein said second data processing cell specification comprises a reserved mnemonic for providing input (sec 7.6.2: \$image-dir ; {size/@width} pg. 37; *item[position]() = \$n*], pg. 49) to the data processing specified by the data processing specification.

As per claim 7, W3C discloses wherein said first data processing cell specification is a reserved output cell specification specifying output of the data processing specified by the data processing specification (refer to *first cell* of claim 1; *xsl: output, xsl: output method* – pg. 7; chp. 16.1, 16.2 pg. 64-68; *xsl:output* pg. 75).

As per claim 8, W3C discloses wherein said second data processing cell specification comprises a conditionally executed formula (e.g. *<xsl: if... />* pg. 74; *<xsl: otherwise ... />*– pg. 75).

As per claims 9-10, W3C discloses wherein said data processing specification further includes one or more global attributes (e.g. *xsl: stylesheet version = "1.0" xmlns:xsl="http://... xmlns="http://www.w3.org/1999... /strict">* pg. 7, 9) specifying one or more global processing characteristics for the specified data processing; wherein said one or more global attributes

include a global attribute specifying a format for providing the specified data processing with an HTTP request (e.g. <xsl: stylesheet version="1.0" xmlns:xsl="http:// ... /strict"> pg. 83).

As per claim 11, W3C discloses an apparatus comprising:

at least one storage unit having stored thereon programming instructions that are configured to be executed by a computer processor and designed to:

receive as input computer code a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation (refer to claim 1) to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell (refer to claim 1);

wherein each cell is delineated by a beginning and ending tag (refer to claim 1), and one of the cells is reserved as an output cell for outputting a result of the processing (refer to claim 1);

parse the specification to determine an interdependency of the plurality of cells and generating and storing a graph of the interdependency as an execution flow (refer to claim 1); and

execute the computer code of the specification in accordance with the execution flow, wherein the executing comprises evaluating the formula of each cell in the execution flow and generating an output result (refer to claim 1);

wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell (refer to claim 1); and

at least one processor coupled to said at least one storage unit to execute (sec 16: XSLT processor, output pg. 64-69) said programming instructions.

W3C does not explicitly disclose graph of interdependency as a directed graph; but this 'directed graph' limitation has been addressed in claim 1.

As per claims 15-20, refer to claims 5-10, respectively.

As per claim 21, W3C discloses computer with a memory having stored thereon instructions that when executed cause to the computer to implement data processing comprising:

means for receiving a data processing specification comprising a plurality of cells, wherein each cell comprises a formula specifying an action or computation (refer to claim 1) to perform when the cell is executed, and one or more attributes referencing other cells, wherein the formula of a first cell may reference a value of a second cell (refer to claim 1);

wherein each cell is delineated by a beginning and ending tag(refer to claim 1), and one of the cells is reserved as an output cell for outputting a result of the processing (refer to claim 1);

means for parsing the specification to determine an interdependency of the plurality of cells and generating and storing a graph of the interdependency (refer to claim 1) as an execution flow; and

means for executing the specification in accordance with the execution flow (refer to claim 1), wherein the executing comprises evaluating the formula of each cell in the execution flow and generating an output result (refer to claim 1);

wherein each cell is interlocked with at least one other cell through the formula or attribute of each cell (refer to claim 1).

W3C does not explicitly disclose graph of interdependency as a directed graph; but this 'directed graph' limitation has been addressed in claim 1.

/*****/

(10) Response to Argument

USC § 103(a) Rejection using W3C

(A) Appellants have submitted that for claim 1, W3C (W3-Xpath and W3C-Xslt) fails to disclose the language of claim 1. More specifically, the argument states that, none of the components of W3C (or precisely W3C-Xslt) processing model, as conveyed by the office action - model including the 'source tree', 'result tree', 'style sheet' -- fulfills “*wherein each cell comprises a formula ... reference a value of a second cell ... wherein each cell is delineated ... wherein each cell is interlocked ... other cell ...*” of independent claim 1 (Appellant's Remarks pg. 11). To further the above observation, the Appellants assert that the XML source tree in W3C (at sec. D.2) for including mere data node cannot include a “formula specifying an action or computation to perform”, nor does the tree include one or more attributes referencing other cells (Appellant's Remarks pg. 12, bottom, pg. 13 top).

Stylesheet (XSL) methodology is a template-based formalism to provide a markup formalism with structure syntax embedding rendering rules that would support proper rendition of a source document (usually a HTTP type document). Source document as exemplified in W3C is usually a markup type of document like a XML having a W3C extensible tree like schema whose a X-Path processing (see W3C-Xpath) would be provided to correlate the source elements interdependencies (node of the XML source document depicted via a Xpath methodology) with their internal data dependencies such that a template (implemented via such XSLT processor) can be formalized to enforce W3C type of document (like XML) rendering as set forth above (*).

The received 'data processing specification' as claimed can be viewed as

(i) any of specification which is result from deriving semantic relationship among nodes of such input XML document being parsed into a tree (which is integral to a interpreting of a extensible markup language hierarchy, using X-Path as in W3C), a specification defined from analyzing this input XML tree or

(ii) a stylesheet specification based on said initial tree specification which is represented by a cell in a XSLT template in the sense that this *template* – which is to be processed by a XSLT processor – is generated to implement the proper rendering (refer to (*)) of such source document based on the data dependencies as presented above.

Hence, the XML as construed from the Office Action (in light of W3C) is merely the remote source including node specification, and corresponding to which an instance of XSL specification is received by a XSLT processor. The XSL specification being processed by a XSL engine as proffered in the Office action as one W3C *template* entails that it was derived from processing specification (based on a source element, a node in the X-path, a embedded specification of a XML); i.e. this derived specification being in form of a cell in a XSLT *template* instantiated to provide this formalism known to support W3C markup presentation rendering. This formalism has structural language (e.g. a XSL template) with tagged content (or tag enclosed cells) implementing data value resolving (or markup content data dependencies) among cells created by way of this XSL formalism. As cited in the Office action, this syntactically structured language enforce rules being applied for proper rendering a given markup document being taught in W3C, structure such as a template having *stylesheet* programmatic constructs (or formula - e.g. *value-of*, *select*) and their underlying operations

(computation), embedded within the very template markup language (embedded within the cells of the template), operations represented by operations or formula, all construed as *attributes referencing other cells*.

That is, the cell of the XSLT template (XSL *template* - see W3C: sec 5.4, sec. 7.6.1) represents the cell among the “plurality of cells” being instantiated based on derivation of data dependencies by some schema type specification received at the source level of the initial XML, a document for which stylesheet presentation is to be made via W3C-XSL methodology. Unlike the allegation by the Appellants, the XML source code is not cited to match the cells of ‘plurality of cells’, but it is the very cells of the template (emphasis added - the template being processed by the XSLT engine) that amounts to ‘plurality of cells’, each having a formula specifying an action, each action (or formula - e.g. *value-of, select*) enforcing markup node/data value resolution to meet the requirement or specification of the target document, which was source for instantiation of a stylesheet template (refer to (*)). As presented in the Office action, the XSLT processor that takes the mapping result correlating a node of a XML tree with cells of a template, then processes the template for syntactic validation of the template content; and this fulfills the ‘receiving’ step recited as ‘receiving as input ... a data processing specification’, because processing cells of a template entails that the XSLT receives a cell for processing, a cell being generated based on an initial specification.

The above analysis can be summarized in that the Office interprets ‘data processing specification’ --- or cell having a formula -- is being received by a XSL processing engine, the “receiving” step of such *data processing specification* -- interpreted as (i) or (ii) -- amounts to a XSLT processor processing a received cell (cell having a formula inside a template)

implemented to correspond to a data requirements obtained from a initial XML tree parsing (W3C and X-path, or DOM).

The claim language does not provide sufficient details to 'data processing specification' in order for the broad language to distinguish from any specification construed as (i) or (ii) as mentioned above. The allegation that XML source nodes (section D.2) are not cells each having a formula is not commensurate with the analysis by which the office correspond template cells having formula such as "select", "value-of" to implement the proper rendering of a source document. Therefore, the argument is deemed non-persuasive.

(B) Appellants have submitted that as proffered in the W3C, a "result tree" (section D.2) amounts to mere fragments, each by applying a template, such that this *result tree* does not include a formula or computation or does not include attributes referencing other cells (Appellant's Remarks pg. 14, top). The rejection has matched 'plurality of cells' with cell inside a XSL template, each cell having a formula, and this has been presented at length in section A. The argument is deemed not commensurate with the language mapping exactly presented in the rejection, hence non-convincing.

(C) Appellants have submitted that the Office's position that W3C's *template rules* disclose 'cell comprises a formula specifying an action ... the formula may reference a value of a second cell' is incorrect, because xsl:template element only includes a match attribute that identifies a source node to which the rule applies (Appellant's Remarks, pg. 15 bottom) since a match attribute is not a "formula" specifying an action (to perform) as claimed.

According to this Appeal's section (5) - Summary of Claimed Subject Matter – the execution of a formula's action is disclosed at pg. 6 of the Specifications, within which a "xcell"

of name “preferences” is defined such to inter-relate with an <output> cell such that this <x:output> cell would retrieve the actual value for the referring xcell whose name (“preferences”) is specified. The inter-relation is such that the actual value obtained via executing operation “value-of select” is stored as \$preferences inside the <output ... /> portion referred to a definition of the very “preferences” variable defined in the upper <x:xsheet> <x:xcell name = ... x/cell> portion. The formula or operation like <value-of select=...> as disclosed is further described in X-cell formulas at page 9 of the instant Disclosure (**); which is indication that X-cell or <Xsheet> is analogous to the xsl:stylesheet cited as <xsl:template ... /> in W3C. For example, W3C section 2.3 shows <xsl: template match header with title “Expense Report Summary”, such that operating a formula <xsl: value-of select = “expense-report .../”> would resolve the inter-locking dependency between the head of the template and the body of the template. Further, section 5.4 of W3C shows template match = “author group” in a header definition to bind the result obtained via executing <xsl: apply-templates select=“author” /> with said template *match* definition defined on top. Clearly, each cell inside a template contain attributes or variable that is referred by another variable, or a formula whose execution yield a actual value bound by a variable defined in a upper cell. Besides, the formulas disclosed by the invention (see **) such as *value-of*, *select* reflect the very operations disclosed by the W3C methodology as “value-of select = ... /> (e.g. sec 2.3 pg. 9) whereas, similarly, the <x:sheet> depicted at page 6 or 7 of the instant Specifications is reminiscent of the structured language of *stylesheet* terminology as cited in the rejection (sec. 5.4 pg 21-22; sec. 7.6.1, pg. 36).

The claim language recited as ‘*cell comprises a formula specifying an action or computation to perform when the cell is executed, and one or more attributes referencing other*

cells wherein the formula of a first cell may reference a value of a second cell has been interpreted as a group of cells having two cells in which one (e.g. a second cell) includes a formula (e.g. select =) that requires action, such that the value obtained therefrom is referred to by a first cell; and the “referencing” as claimed --by way of attributes defined in one cell – amounts to defining a variable in the first cell which would interlock or bind an operation to be executed by way of a formula included in the second cell, as set forth above.

Revisiting section (5) of the Brief, it is shown that what started at pg. 5, lines 11-12 of the Specifications, to represent ‘formula of a first cell may reference a value of a second cell’ is further explained with illustration at page 6.

The ‘may reference a value of a second cell’ is therein disclosed as formula containing a reference to a location holding value of a variable defined in a second cell. Specifically, page 6 of the Specifications shows that a *first cell* has

“value-of select = “\$preferences/mydata/favoritecolor”/”>

where this formula “value-of select” (Specs, pg. 6 lines 22-23) requires action ‘select’ to be performed”, such that the resulting value is found at a location “\$preferences/./favoritecolor”, location referred to as \$ operator, a reference that can be a *directory path* with multiple “/” (forward-slash) and which contains the obtained actual value for variable ‘preferences’ defined in another cell (i.e. second cell - see Specs, pg. 6 lines 15) different from the first cell as mentioned above. Also at page 6 of the Specifications, the other *second cell* defines <x:xcell name = “preferences”> whose actual value is retrieved by a reference to location = “\$preferences/./favoritecolor” by way of operation ‘select’ included in formula “value-of select” of said first cell. The Specifications teaches that ‘select’ and ‘value-of’ are both

formulas(Specifications pt. 9) where \$ is a form of mnemonic to indicate location at which a value obtained for the operation can be stored.

The binding (interlocking) of first cell to a second cell via defining a variable, in the first cell and obtaining the value thereof in another cell is clear based on the very teaching of the Disclosure. The first cell defines a variable whose actual value has to be obtained by processing the formula in the second cell, the second cell's formula referencing location to retrieve such value.

W3C also teaches a (second) cell defining a variable and a separate (first) cell including 'value-of select=' with reference to a location or pointed to variable (e.g. a path) that contains the actual value for the variable defined in the other cell. Examples in W3C can be given as follows.

Sec 11.2: <xsl:variable name="n" .../>

<xsl:value-of select="item[\$n]" /> (pg. 50 top);

Sec 7.6.1: <xsl:template match = 'person'>

< xsl:value-of select="@given-name" />

<xsl:value-of select="@family-name" /> (pg. 37 top)

Sec 7.5: <xsl:template match=@* | node() ">

<xsl: apply-template select="@*| node()" /> (pg. 35)

Sec 5.4: <xsl: template match="author-group">

<xsl: apply-templates select="author/given-name"/> (pg. 22 bottom)

Sec 11.4: <xsl:template match="para">

<block font-size = "{para-font-size}"> (pg. 51- note: the formula being *font-*

size)

The template match (by W3C) is illustration of a way to validate portions of a document (HTTP, XML) for display at a browser, validation done by apply rules of the template, in which 'select' included in one cell can retrieve the value for a variable called for in another part (another cell) of the template for match.

W3C template and 'select' formula to reference a value stored at a location variable meets the language of 'formula in a first cell' that may 'reference a value of a second cell'.

The claim language thus recited does not remotely include any restrictive language that would preclude the matching scenario as cited in the office action; that is, a template matching which requires application of "value-of select" (and result thereof) from one part of the template (a second cell) to a variable defined (first cell) in another part of a <xsl:template> in W3C, because the resolution of a variable whose name is defined in a <template match> necessarily binds operation result construed in a formula (value-of select) defined in a second cell part of the template, as this is the basis of matching is W3C or XSL language, which appears to be the very operation (and referencing of attributes between upper and lower cells) to yield a result described in the instant Disclosure (pg. 6, lines 14-25).

The "formula" and the 'referencing' as claimed (see claim 1) are therefore matched by W3C; i.e. the above Appellants' argument inter-relating result tree and template is hence insufficient to overcome the rejection.

(D) Appellants have submitted that 'match' attribute and/or 'select' attribute in a template rule by W3C only matches pattern of a node (representative of part of an expression) to a rule, hence cannot teach 'formula in a first cell' references a value in a second cell (Appellant's Remarks pg. 16, top pg. 17). The claimed 'formula ... references a value in a second cell' has

been interpreted based on the operation explicitly explained in the Disclosure and again enclosed in the Summary (of subject matter) as per **section (5)** of this Brief. This operation implicates definition of one variable in an upper cell whose value is to be obtained from executing a formula defined in a second cell as the operation to be executed by a processor, the programmatic definition of this operation referencing a value to be retrieved and that relates to a variable defined in the upper cell. The executed operation based on a formula and the related context for *referencing a value* as understood based on the Specifications has been matched with W3C as described in section C. Again, the claim does not provide details that would dictate constraints that otherwise would prohibit use of a template *select* formula in one XSL cell, by way of its execution, to match value referred to by any other variable in another XSL cell, as is the case of W3C template-match. The argument is therefore non-persuasive.

(E) Appellants have submitted that “xsl:value-of” as cited is used to extract source text and insert variable value as shown in sec 7.6.1 of W3C; and this is not same as ‘formula of a first cell referencing a value of a second cell’ (Appellant’s Remarks, pg. 17 middle, pg. 18 top). Definition of a formula as in W3C includes reference to a value such as @given-name and this value is associated with its programmatic representation of variable defined in the another cell such as <match = “person”>, and the operation, included in one cell, to resolve the *value-of* (referencing) operator therein is to be executed via “select=” formula whose result is stored in location @given-name or @family-name, a value that would resolve its interlocking relationship with variable “person” defined in another cell. The claim language in its state, and in view of the clear teaching by the Invention’s Disclosure, is deemed not specific to prohibit the template match and use of ‘value-of select=’ as cited by the office action. Appellant’s arguments fail to

comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

Further, a quick look as how 'value of' element is depicted in the instant Disclosure reveals (see Specifications: pg 10, middle) that a variable result from executing "select" formula is based on this referencing operator "value-of" so that the output value is "JohnDoe", evaluation shown as `<x:test ... actual <x:value of select=$xxx /> being` processed along the path of a evaluator program so to obtain corresponding values (see Specifications, pg. 12) needed for *document fragment result* (emphasis added) result of said test (`x:test`), all of which not far distinguishing from W3C's applying of the same `<value-of select=@given-name>` in sec 7.6.1 as mentioned above, the obtaining of result to meet requirements of a template to validate data inside fragment of a document.

(F) Appellants have submitted that W3C fail to teach or suggest reserved cell as an "output cell" because W3C outputs result tree portions based on applying template just is an incorrect matching of the recited "reserved cell" (Appellant's Remarks pg. 18). The claim language has been given merits using broad reasonable interpretation. The 'output result' as claimed is viewed as a result or output from evaluation of formula of cells being processed as *execution flow*, the output as a result being from executing the flow including the evaluation, and stored with a dedicated cell, which is cited as `<xsl:output>` in W3C. The claim does not relate "output" reserved cell to the formula in terms of its being a reserved `</...>` type container specifically created to collect the value obtained from the action based on formula. The claim only describes this cell as to one cell to obtain result of the processing, the processing construed based from **a)**

the “execution flow” recited in the claim and **b)** the disclosed evaluating process described as sequences of validating <value-of select= ...> or execution flow of a evaluator in order to match a require fragment of document (instant Disclosure: *document fragment result* - Specifications, pg. 12, bottom) by retrieving the values dictated by the formula specification of the <x: cell>, which is reminiscent of the fragment of result tree for a given part of a document being matched with the requirement of a template as in W3C, the result output being shown as collected in a particular <output> cell. One cannot see how using a XSLT processor as in W3C to match a node specification with a (value-of, select) formula enclosed in a <match> template to output a proper fragment (result tree) would clearly distinguish from processing sequences of ‘value-of ... select ...’ formula to validate a given document output as by the instant Disclosure (emphasis added). The “output cell” for outputting a result in light of a) and b) has been properly matched with W3C. Appellant’s arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

(G) Appellants have submitted that “<xsl:value-of” element which is used to generate text as construed from sec 7.6.1 cannot be same as a formula in the sense that “each cell is interlocked with at least another cell through the formula” or attribute (Appellant’s Remarks pg. 19, top para). The tight relationship (interlocking effect) between ‘select=’ in view of the referencing operator “value-of” in order for proper value validation of a target document fragment specification required from matching XSL template has been discussed above in sections C, D, and E. The argument is largely insufficient to preclude the template-matching process and result tree fragment generating in W3C.

(H) Appellants have submitted that “mixes and matches” was used by the Office action in reasonably articulate the rationale to meet independent claim 1, particularly in the context that W3C 3 discrete components as *source tree*, *a style sheet*, and *a result tree* forming XSLT model cannot be combined to reach the features of claim 1 (Appellant's Remarks pg. 20). What appears to be a global/aggregate allegation for patentability in associating a XSLT model with features of the claim, as from above, does not seem parallel with the very cited W3C sections provided by the Office Action, sections specifically matched with each limitation of the claim, as the features of claim has been interpreted and whose rejection has been explained in the above sections. One cannot see non-obviousness of the features of the claim 1 when there is no factual rebut (e.g. where does a cited section in W3C differ from a formula? from a plurality of cells) addressing specific portions of the Office action in relation to each and all the features of claim 1.

(I) Appellants have submitted the office action fails to disclose or suggest plurality of cells, each delineated by beginning and end tags, each having a formula (Appellant's Remarks pg. 21, top) and enforcing a interlocking effect among the cells as required from claim 1, in view of the lack of articulate reasoning by the Office Action. The above allegation is to be referred to the Response sections addressing claim 1 from above.

(J) Appellants have submitted that ‘xsl: value-of’ as cited in W3C cannot teach ‘first cell has a first attribute referencing a second attribute of ... second cell’ (re claim 5 – see Appellant's Remarks pg. 22). The relationship between cells in light of a formula or referencing a attribute in one cell by another cell is construed based on the Disclosure, specifically as portions where <value of> attribute and “select” formula is taught, all of which mentioned above as representing the claimed ‘formula’ and ‘referencing’ of claim 1. Hence, the response to address Appellant's

argument that W3C 'value-of' cannot match 'referencing' (claim 1, claim 5) as set forth above herein apply.

(K) Appellants have submitted that (for claim 6) W3C fails to disclose 'reserved mnemonic for providing input to the data processing specified by the data processing specification', because W3C template attribute makes no mention of 'mnemonic' (Applicant's Remarks pg. 22). An abbreviated form of syntax that facilitate reuse so not to burden code implementation with too much verbose is considered mnemonic, since the claim does not provide more details (emphasis added) as to what exactly this limitation amounts to. Programmatic construct as shown within a W3C template uses a short code for a location to store a value (sec 7.6.2: \$image-dir ; {size/@width} pg. 37; *item[position() = \$n]*, pg. 49), this short code form being a \$, @ , a program symbol/operator viewed as an abbreviated construct to obviate verbose code form. The short symbol, e.g. "\$" being integral to the XSL programming syntax, therefore reasonably conveys the code implementation concept of "mnemonic" for representing a place for inputting a result or a value. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

According to the instant Specifications, a "\$" is mnemonic for representation of location at which to retrieve value of operation like 'select' 'value-of' (see *for providing input* - Specifications, pg. 9)

Accordingly, W3C reasonably disclose this 'providing input' limitation, and example of using this same mnemonic "\$" in terms of a location designed to take in a value is provided in

the action to retrieve font-size in W3C, where the \$xxxx is a location variable for providing a input place for result of a “font-size” operation, as follows:

Sec 11.4: <xsl:template match=“para”>

<block **font-size** = “{\$para-font-size}”> (W3C, pg. 51)

The argument is therefore insufficient to overcome the rejection of claim 6.

(L) Appellants have submitted that the ‘first cell’ being reserved for specifying output of the data processing in claim 7 cannot be same as ‘xsl: output’ featured in W3C, where the W3C output cell is an optional action to output fragment of a result tree (Appellant's Remarks, pg. 23).

Claim 7 includes two instances of output cell: one as a reserved to include output from executing a flow including evaluation of cell formula, the other being reserved first cell specifying output, the first cell with formula that may reference a value of a second cell. Addressing the ‘first cell formula’ specifying an output, the *output cell* and embedded *output method* defined in sec 16.1, 16.2 reasonably matches the reserved cell recited as *first cell having a specification (or formula or method)* that internally *references an value* (as in claim 1) and outputting it, output resulting from tree node or variable processing via template matching, e.g. validating actual value associated with defined variable in *second cell* of a template. A special cell with *output method* **reads on** a cell having a specification that may reference an output value obtained from another cell, and this fulfills claim 7, based on broad interpretation as follows.

From claim 1, it is understood that each cell among the plurality thereof comprises one or more attributes referencing other cells or can be reserved for outputting processing result, one such cell (first cell) can have a formula that may reference a value in a second cell. Hence, the very construct for outputting defined within as a reserved <xsl: output> cell in W3C (xsl:output

... method = ...) explicitly satisfies *reserved cell* having a method specification or programmatic *formula* referencing a value in that it implicitly retrieves a proper value then sends it as output (e.g. a fragment of XML document), the value (e.g. value obtained at @given-name) of a variable defined in another cell, value being resolved by matching using W3C XSL evaluation engine effectuated on plurality of templates to yield a validated output fragment. The output “method” (sec 16 of W3C) therefore reasonably fulfills a specification that references a value in a second cell, the specification such as a formula embedded in a reserved cell (e.g. the <xls:output>) the cell operating with a ‘method’ for getting a value like a fragment of a document or part of result tree, the document whose text (or variable actual value) are being validated by the XSLT execution of template-matching using the requirements inside template cells.

Further, as shown in the Office action, a reserved cell specification specifying output of the data processing can be viewed as a cell within a template having a select = @given-name as formula (see W3C: sec 7.6.1) , the formula when executed may reference an address location (refer to section E from above) where a value is stored (*formula that may reference a value*) as a output resulting from 'data processing specification' or template-matching cell with ‘value-of’ referencing and “select =” as formula as addressed in claim 1. Hence, the very cell inside a template-mach can also **read on** a first cell implemented as reserved cell with formula that may reference a value (value which is broadly construed as the actual value of the associated variable in another second cell different from the first cell – i.e. *value in a second cell*) in light of the instant Disclosure supported at pg. 6 (*value-of select=\$preferences* - lines 6-25) and the Brief’s Summary of Claimed Subject Matter (Appellant’s Remarks pg. 6, top para - emphasis added).

Based on the above, claim 7 is deemed met by the Office Action.

(M) Appellants have submitted (for claim 9) that W3C fails to disclose or suggest 'one or more global attributes ... global processing characteristics for ... data processing' because version attribute in "xsl:stylesheet version" is a mere version required by the XSLT, which is not 'global attributes specifying ... processing characteristics ...' (Appellant's Remarks, pg. 25). The claimed 'one or more global processing characteristics' is not construed as containing a particular restrictive constraint that would significantly read away from the XSLT global attribute such as a version characteristic used to dictate a particular engine to process the language protocol. The argument for merely alleging on patentability of a broadly claimed language/feature is deemed insufficient to overcome the rejection.

(N) Appellants have submitted that 'global attribute specifying a format ... processing with a HTTP request' as required in claim 10, is not fulfilled by the namespace attribute cited as "xmlns:xsl" (Appellant's Remarks pg. 26). The use of stylesheet methodology as a validating protocol for presenting HTML document is disclosed in the very matching approach of W3C, and when one global attributes specifies a namespace having all the included metadata, support document/files and libraries relevant to using this W3C stylesheet (presentation) protocol or validating mechanism, that globally defined attribute reads on specifying a format (XSL type of format) for providing "specified data processing" with a HTTP request (e.g. for rendering a document). The argument is deemed non-persuasive to prove clear patentability to claim 10.

(O) Appellants have submitted that for claims 11 (Applicant's Remarks pg. 27), claim 15, claim 16 (Applicant's Remarks pg. 28), claims 17-19 (Applicant's Remarks pg. 29-30), Appellants' reasoning discussed in Section VII, A, items *i*, *ii*, *iii*, *iv*, *vi* can respectively apply

because none of the features of the respective claims is disclosed by W3C. This global assertion is referred back to the response addressing each of the above VII-A sections i (claim 1), ii (claim 5), iii (claim 6), iv (claim 7), and vi (claim 9)

(P) Appellants have submitted that the features of claim 21 fall under the reasoning by Appellants discussed in Section VII, A, i relevant to the features of claim 1 (Applicant's Remarks pg. 32). The above reasoning has been addressed in the corresponding section from above.

In all, the arguments are deemed insufficient to prove distinction between W3C and the claim language, in view of the interpretation of the claimed features construed under the very teaching (of relevance) gathered from the instant disclosure; the claims will stand rejected as set forth in the last Office Action.

/*****/

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

Art Unit: 2193

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/ Tuan A Vu /

Conferees:

/Lewis A. Bullock, Jr./

Supervisory Patent Examiner, Art Unit 2193

/Emerson C Puente/

Supervisory Patent Examiner, Art Unit 2196